Zooming while Tracking using Affine Transfer

E Hayman, I D Reid and D W Murray
Department of Engineering Science, University of Oxford,
Parks Road, Oxford, OX1 3PJ, U.K.
Email [ian,dwm]@robots.ox.ac.uk

Abstract

Zoom interacts strongly with both vision and control processes in an active visual system, causing problems for many commonly used tracking methods. This paper demonstrates the use of affine transfer to track while zooming, using clusters of corner features. Affine transfer not only is fundamentally invariant to zoom but also provides a natural mechanism to allow features to appear and disappear while tracking, events which will occur as detail sharpens and dissolves during zooming. The paper demonstrates offline 3D affine transfer during zoom for objects undergoing substantial rotation, and describes real-time experiments using 2D affine transfer while zooming and tracking using an active camera platform.

1 Introduction

Two low-level requirements in visual surveillance are the ability to track and the ability to zoom — abilities which win sufficient time and sufficient definition for higher level recognition processes to function. A variety of general image features suggest themselves as candidates to be exploited for the tracking and zooming process, and their associated methods can be grouped into the three broad categories of region-, contour- and point-based.

For tracking alone, region-based methods, typified by correlation, suffer from the problem that they are not view-point invariant, and that they offer little immunity to local occlusions. Contour methods fare considerably better with respect to view-point invariance and occlusion insensitivity, but at the cost of incorporating prior templates. The ideal method then would appear to be point-based. An image corner feature is view-point invariant, simple to extract and requires no prior model. However, as noted in [11, 12], although clusters of corners exhibit temporal coherence and longevity, individual corners are emphemeral and quite unsuitable for tracking over extended periods.

When zoom is introduced, the situation becomes very much more difficult for all categories of method. Correlation now suffers particular badly because the method is fundamentally not invariant to zoom. Contour tracking, where the template is constrained to deform affinely, is invariant to zoom, but there is the practical problem of what happens if the single contour falls off the image while zooming-in, as would happen to a "whole-person contour" in Figure 1, or becomes very small while zooming-out, as would happen to a "head contour" taking Figure 1 in reverse. Again, a point-based method would seem the ideal. Under active fixation, the single point would be always be near the image centre and hence

BMVC 1996 doi:10.5244/C.10.21

Figure 1: Zooming sufficiently for identification introduces such changes in the image that the performance methods of tracking must suffer, especially those based on 2D models.

could not fall off the image. However, the lifespan of the individual point may be reduced by changes in scale. If one turned to naive methods of clustering several points one would suffer the same difficults as the contour method when corners fall off or enter the sides of image during zooming in and out, and appear or disappear because of scale effects.

Recent work has demonstrated the active tracking of clusters of corner features using the method of affine transfer, both monocularly [11, 12] and stereoscopically [4]. The method finesses the difficulty caused by the temporal instability of a single corner by, in the simplest case of 3D transfer, replacing the requirement to track *one* corner through *all* image frames by the less demanding requirement to track *any four* points across *three successive* frames.

Because scaling is an affine transformation, the method is *fundamentally* invariant to zoom. Moreover, because the method allows corners to disappear and appear, and because the gaze point is not tied to a physical feature, it appears to solve the other problems introduced by zooming.

We demonstrate the method with experiments on 3D transfer using code running offline on a workstation, and on 2D transfer using real-time code where the resultant tracking position is used to control the gaze direction of an active camera platform.

Section 2 reviews the theory of affine transfer. Sections 3 and 4 describe the implementations and give results from the offline and real-time experiments respectively. The results are discussed in Section 5, and conclusions drawn and future work remarked on in Section 6.

2 Theory

The affine transfer algorithm derived from work [6, 10, 5, 3] which showed that structure can be recovered up to a 3D global linear transformation (affine or projective). Such recovered structure is sufficient to compute images from arbitrary novel viewpoints, a process known as *transfer*.

Where scene relief is small in comparison with depth, it is valid to assume an affine camera projection

$$\mathbf{x} = M\mathbf{X} + \mathbf{t} \tag{1}$$

where $\mathbf{x}$ is a 2×1 image position vector, M is a 2×3 matrix, $\mathbf{X}$ is a 3×1 world position vector, and $\mathbf{t}$ is a 2×1 translation vector. Consider a set of four points, $\mathbf{O}, \mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, in general position (non-coplanar) on an object (see Figure 2). The four points define a basis set $\{\mathbf{A}-\mathbf{O}, \mathbf{B}-\mathbf{O}, \mathbf{C}-\mathbf{O}\}$, relative to which coordinates for any point on the object (or, for that matter any point in the world), $\mathbf{X}$, may be uniquely defined by three *affine coordinates*, α, β, γ:

$$\mathbf{X} = \alpha(\mathbf{A}-\mathbf{O}) + \beta(\mathbf{B}-\mathbf{O}) + \gamma(\mathbf{C}-\mathbf{O}) + \mathbf{O} \tag{2}$$

These coordinates are invariant to the affine projection in the sense that the projected coordinates of the point $\mathbf{X}$ are the same linear combination of the projected basis vectors:

$$\mathbf{x} = \alpha(\mathbf{a}-\mathbf{o}) + \beta(\mathbf{b}-\mathbf{o}) + \gamma(\mathbf{c}-\mathbf{o}) + \mathbf{o} \tag{3}$$

Given two views of the four basis points ($\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{o}$ and $\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{o}'$), we can compute the affine coordinates of the fifth point, $\mathbf{X}$ in the two views by solving the over-constrained system of linear equations

$$\begin{bmatrix} \mathbf{x}-\mathbf{o} \\ \mathbf{x}'-\mathbf{o} \end{bmatrix} = \begin{bmatrix} \mathbf{a}-\mathbf{o} & \mathbf{b}-\mathbf{o} & \mathbf{c}-\mathbf{o} \\ \mathbf{a}'-\mathbf{o}' & \mathbf{b}'-\mathbf{o}' & \mathbf{c}'-\mathbf{o}' \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{4}$$

for α, β and γ.

Having computed the affine coordinates of the point $\mathbf{X}$, it is trivial to determine its projection in a novel view, given the projected positions of the reference (basis) points in the novel view, as

$$\mathbf{x}'' = \alpha(\mathbf{a}''-\mathbf{o}'') + \beta(\mathbf{b}''-\mathbf{o}'') + \gamma(\mathbf{c}''-\mathbf{o}'') + \mathbf{o}'' \tag{5}$$

Suppose that while tracking an object undergoing a linear transformation (more general than a rigid one), the desired fixation point was $\mathbf{g}$ in frame t and $\mathbf{g}'$ in frame t'. We can compute its affine coordinates $[\alpha_g, \beta_g, \gamma_g]^\top$ using equation 4 above. Then in frame t'', a short time later, the positions of the four basis points project to new positions, $\mathbf{a}'', \mathbf{b}'', \mathbf{c}''$ and $\mathbf{o}''$, and equation 5 gives a position for the desired fixation point $\mathbf{g}''$ in the new frame. Note that neither $\mathbf{G}$, nor its projections $\mathbf{g}, \mathbf{g}', \mathbf{g}''$ need correspond to a physical feature. Thus with *any* four corner correspondences (in general position) in three frames we can reconstruct the position of the desired fixation point given its image coordinates in the first two frames. The four corners used need not be the same over time: rather, there must merely be *one* set of four corner correspondences between each set of three consecutive frames, as shown in Figure 2(b).

If the no-coplanarity condition is not satisfied, i.e. the scene is planar, only three basis points ($\mathbf{O}$,$\mathbf{A}$,$\mathbf{B}$, say) are required across two views to provide transfer, provided of course, the three points are not collinear.

2.1 Using all the features

To increase robustness, it has been shown [12] that all points can be used, in an algorithm which can be formulated as the factorization method proposed by Tomasi

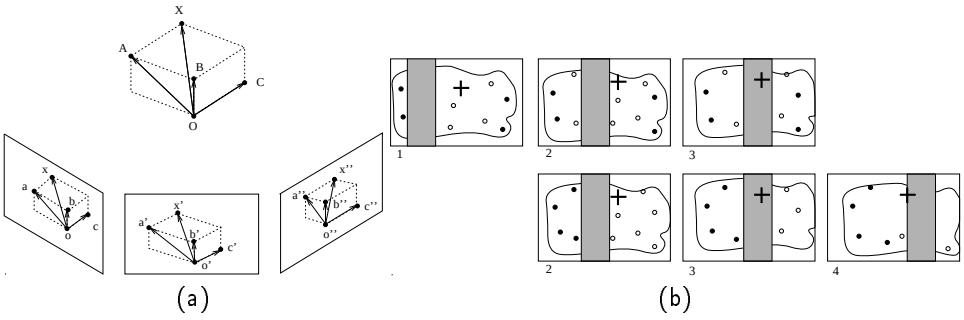

Figure 2: (a) Affine transfer. (b) The basis set for frames 1,2,3 (the black filled points) differs from that of 2,3,4. Note too that the fixation point + is virtual and can be found even when there is partial occlusion.

and Kanade [14], to yield a least-squares estimate for (α, β, γ). (Although in their work they were originally concerned with solving the structure-from-motion problem under orthographic projection and for Euclidean structure, the concepts are equally valid for affine projection and affine structure.) In practice for transfer the full singular value decomposition used in the factorization need not be computed. The timing then depends on the number of views but not points, giving a constant time transfer which is of considerable benefit in the context of real-time tracking. This method is applicable to both 3D and 2D problems, and moreover the singular values indicate if either the 3D or 2D method is approaching their respective degenerate configurations.

More recently, robust methods have been used to eliminate outliers from the transfer computation [8], but here we use the least squares approach.

2.2 Zoom invariance of affine transfer

It is evident from the above that the computation of the affine coordinates does not depend on the specific M and $\mathbf{t}$. The only requirement is that each image used is related to the scene by an affine transform. Indeed, the scene itself need not be rigid, but could undergo an affine deformation.

To a first approximation, zooming the lens simply scales the image preserving the affine nature of the transformation. Note that the invariance to zoom implies that no knowledge of zoom is required, and so the method is free of calibration.

Although affine transfer is zoom invariant, other difficulties are introduced for an active system, and these are returned to in the discussion.

3 Offline implementation

3.1 Method

For the offline experiments the algorithms for both 2D and 3D transfer using all points was coded in C on a workstation using the Horatio vision libraries [7]. Image corners were detected using the Plessey corner detector, and correspondence established using a simple matcher. The object/background segmentation problem

was eliminated artificially by setting a window over the object, and keeping the window to a fixed size while zooming in. (This method often has the unfortunate side-effect of reducing the amount of data collected.) A 8.5–42.5mm zoom lens as used with a 8.8×6.6mm format CCD camera, giving angular fields-of-view from 50–10°. The motorized zoom was operated open-loop, whilst viewing a moving object. We note that for the 3D method to function, sufficiently different views must be obtained so that the structure can be recovered. Whilst this is achieved easily using stereo [4], using a single fixed camera requires object rotation (or camera rotation if visually servoing). There is of course no such requirement in the 2D algorithm used in the real time work.

3.2 Results

We show results from an experiment where the subject rotates his head while the camera zooms in. Corner features were detected within the window set initially to cover the face. The fixation point is set to the centre of the window in the first frame. In subsequent frames the order is reversed: the gaze point is calculated using 3D transfer, and the detection window then moved.

Figure 3: Tracking offline while zooming onto a rotating face. Corner features were detected across the face and the fixation point set initially to the middle of the detection window.

4 Real-time implementation

4.1 Method

In the real-time experiments the zoom lens and camera were mounted on a mechanical stereo platform using only the elevation axis (up-down) and one vergence axis (left-right). The axes, driven by geared DC servo motors are capable of accelerations of up to $6000°s^{-2}$ and minimum/maximum speeds of $0.03/400°s^{-1}$. The control sub-system, running on a single T805 transputer consists of two parts: a low level servo controller, running at 500Hz, receives feedback from the motor shaft encoders and generates appropriate motor torques, and a higher level part which operates asynchronously as determined by the visual processing and selects visual output to drive gaze constructs such as pursuit (tracking) and saccades, interpolating the visual demands up to a synchronous 500Hz demand for the servo controller. The visuo control scheme is sketched in Figure 4.

Corner detection and tracking were implemented on a group of three transputers (two for corner detection, one for matching and tracking), which divide a 64×32 central (or "foveal") window spatially and process at 25Hz with a latency of less than 70ms. For speed we used the detector of Wang and Brady [15, 1], rather than the Plessey detector used offline. Spatio-temporal correspondence of corners was achieved using simple variation [12] of the algorithms proposed in [2, 13]. An important feature of the variant is that the image motion of the corners induced by the movement of the camera is calculated for each frame using odometric information and subtracted from that observed.

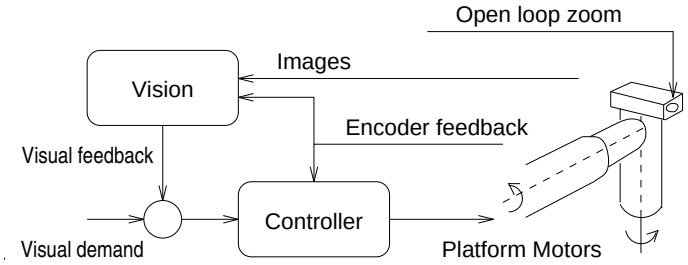

Figure 4: The visuo-control loop. Encoder data is fedback not only to the servo-controller, but also to the vision module to allow subtraction of induced motion. The zoom lens is run open loop.

4.2 Results

We first show results from the basic real-time 2D affine transfer implementation; that is, without using all points, without zoom, and without controlling the camera. It shows more clearly the tracked corners and the changing basis set.

Several frames cut from a video taken through the lens of the active camera are shown in Figure 6. The buggy is reversing to the left, and the camera is fixated on the front mudguard.

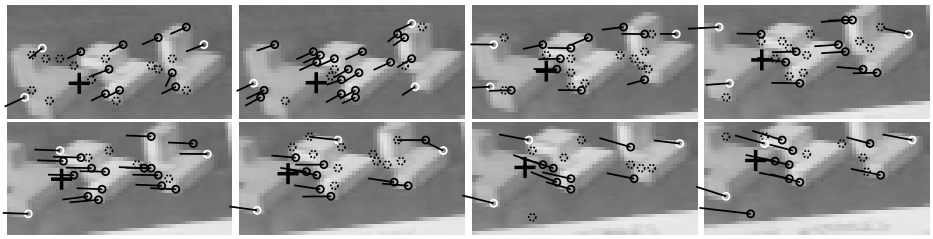

Figure 5: Defining a gaze direction using 2D affine structure: Dotted circles show the positions of unmatched corners, solid circles show the positions of matched corners, each with a velocity vector, white solid circles indicate the current basis set and a cross-hair indicates the desired fixation point. (Every fourth frame is shown from a 1120ms sequence.)

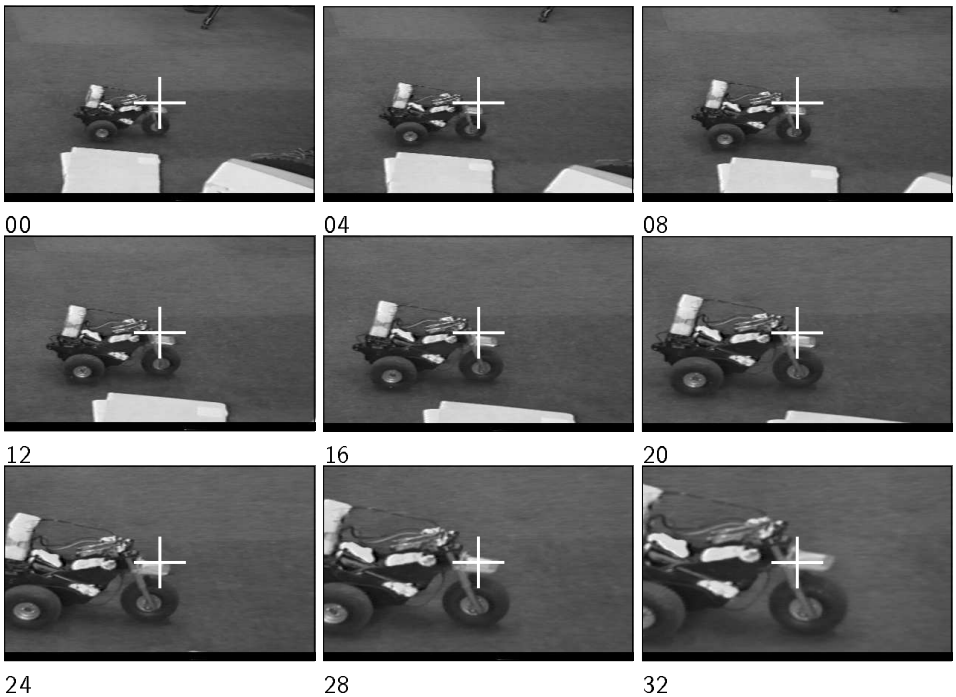

Figure 6: Real time results captured through the lens of the active camera.

5 Discussion

Although our experiments have demonstrated the feasibility of zooming while tracking, there are outstanding issues which affect the stability of the process.

Three such issues arise from the changing relationship between pixel distance on the image and gaze angle of the camera as the focal length of the lens changes. First, when zooming in upon a target which has constant velocity in the scene, the image velocity increases, affecting the performance of the real time corner matcher. Remedying this would require search parameters that vary using an estimate the focal length. Secondly, as noted in [9], although the focal length of the camera is strictly not required to obtain fixation in an active system which uses feedback, the response of the system does depend on it. Given an error Δx on the image plane, the angular error is $\Delta\theta = \tan^{-1}(x/f)$. Thus if f is overestimated, the angular error is underestimated, and the platform will respond sluggishly and appear overdamped. Conversely if f is underestimated, the system will appear underdamped. In our work we set the focal length to be that at mid-zoom, and thus the system changes from over- to under-damped as the lens zooms in. Again, a continuous estimate of the focal length is required to correct this. Thirdly, the image motion induced by rotation of the active camera depends on focal length, and the ability to segment the independently moving foreground from a static background will deteriorate.

A further issue which remains unexplored is that of scale-space effects during zooming. This must affect the localization of corners, and how new corners appear as the level of details increases. Both of these effects in turn are altered by defocussing which occurs during zooming.

The frames shown in Figure 7 are typical of the failure of tracking at full zoom.

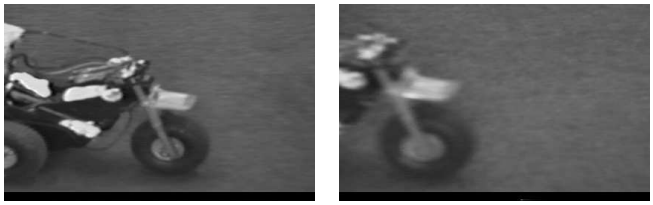

Figure 7: Tracking fails at high zoom. The most likely cause here is that the image velocity has increased to the point where the point matcher fails. However, defocussing and motion blur may also be contributing.

6 Conclusions

We have shown experimentally that the method of affine transfer for tracking clusters of features is indeed invariant to zooming of the camera lens. 3D affine transfer during zoom has been demonstrated offline for objects undergoing substantial rotation, and results for a rotating face given here. Real-time experiments have demonstrated 2D affine transfer during zoom, and the tracking error used to control an active camera platform. Although the method fundamentally needs

no calibration, the changing focal length interacts both with the visual processing and the control processing, making some estimate desirable.

We remark that zoom is a process which is unique in that it interacts with every aspect of visuo-control, and generates difficulties for all categories of feature detector. Our initial studies show that the use of affine transfer with clusters of point-based features goes a considerable way to mitigating its affects, but further study, particularly of the way that zoom affects low-level detection, localization and matching, is required before zooming over the range of Figure 1 can be handled routinely and robustly. Finally, again referring to Figure 1, there are unaddressed questions of how to zoom onto the most interesting part of an tracked object — the face rather than the feet, say.

Acknowledgements

This work was supported by Grant GR/J65372 from the EPSRC, and by a Glasstone Fellowship from the University of Oxford to IDR.

References

[1] J. M. Brady and H. Wang. Vision for mobile robots. *Phil. Trans. R. Soc. Lond. B*, 337:341–350, 1992.

[2] J. M. Brady, H. Wang, and L. Shapiro. Video-rate detection and tracking of coplanar objects for visual navigation. In *Proc. 2nd Int'l Conf. Automation, Robotics and Computer Vision*, 1992.

[3] S. Demey, A. Zisserman, and P. Beardsley. Affine and projective structure from motion. In D. Hogg and R. Boyle, editors, *Proc. 3rd British Machine Vision Conf., Leeds*, pages 49–58. Springer-Verlag, September 1992.

[4] S.M. Fairley, I.D. Reid, and D.W. Murray. Transfer of fixation for an active stereo platform via affine structure recovery. In *Proc. 5th Int'l Conf. on Computer Vision, Boston*, pages 1100–1105. IEEE Computer Society Press, 1995.

[5] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Proc. 2nd European Conf. on Computer Vision, Santa Margharita Ligure, Italy*, pages 563–578. Springer-Verlag, 1992.

[6] J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *J. Opt. Soc. Am. A*, 8(2):377–385, 1991.

[7] P. F. McLauchlan. Horatio: libraries for vision applications. Technical Report OUEL 1967/92 (revised), Dept. Engineering Science, University of Oxford, November 1994.

[8] P F McLauchlan. Real-time 3d affine reconstruction with an active fixating camera. Private Communication, 1996.

[9] P.F. McLauchlan and D.W. Murray. Active camera calibration for a Head-Eye platform using the variable State-Dimension filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):15–22, 1996.

[10] L. Quan and R. Mohr. Towards structure from motion for linear features through reference points. In *Proc. IEEE Workshop on Visual Motion*, 1991.

[11] I. D. Reid and D. W. Murray. Tracking foveated corner clusters using affine structure. In *Proc. 4th Int'l Conf. on Computer Vision, Berlin*, pages 76–83, Los Alamitos, CA, 1993. IEEE Computer Society Press.

[12] I. D. Reid and D. W. Murray. Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision*, 18(1):1–20, April 1996.

[13] L. S. Shapiro, H. Wang, and J. M. Brady. A matching and tracking strategy for independently moving objects. In D. Hogg and R. Boyle, editors, *Proc. 3rd British Machine Vision Conf., Leeds*, pages 306–315. Springer-Verlag, September 1992.

[14] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[15] H. Wang and J. M. Brady. Corner detection for 3D vision using array processors. In *Proc. BARNAIMAGE-91, Barcelona*. Springer-Verlag, 1991.